Design :

The basics of both program first includes reading the input file and noting down in to arrays actually a beteer way would be use of lists but simmilar results can be obtained by arrays so I confined to the arrays

RMS:

Rate monotonic scheduling is giving priority based on priority execution times

- It is static-priority scheduling
- the priorties don't change whith time

Generally we don't choose rms if we are missing deadline in any formatbut the question say let us assume that to do it even if it misses deadline

This bring a chioatic situation of assigning cpu schedule to the process whic is going to miss the deadline.

The assumptions Here are :

- If a scheduler knows if it starts the process and run it without intreputionand still the process misses the deadline then it will not assign the cpu time to that process
- Else it will be excuted

Based on this assumption at every second I checked for the next comming process and reduced the remaing exec time of that process if there is the change in next process I updated the log file

EDF:

Earliest Deadline first this is a bit easier than the above.

In this at every instance we schedule process based on earliest deadline

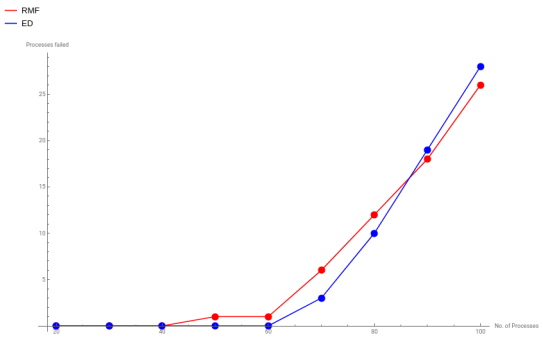
Conclusion :

Both Algorithms has there disadvantages as edf may give more priority to deadline even the process is slow and rms doesn't consider deadline and it is static these

both were not optimal ways of scheduling but they do have their roles in many specific locations and modern one's mix them both

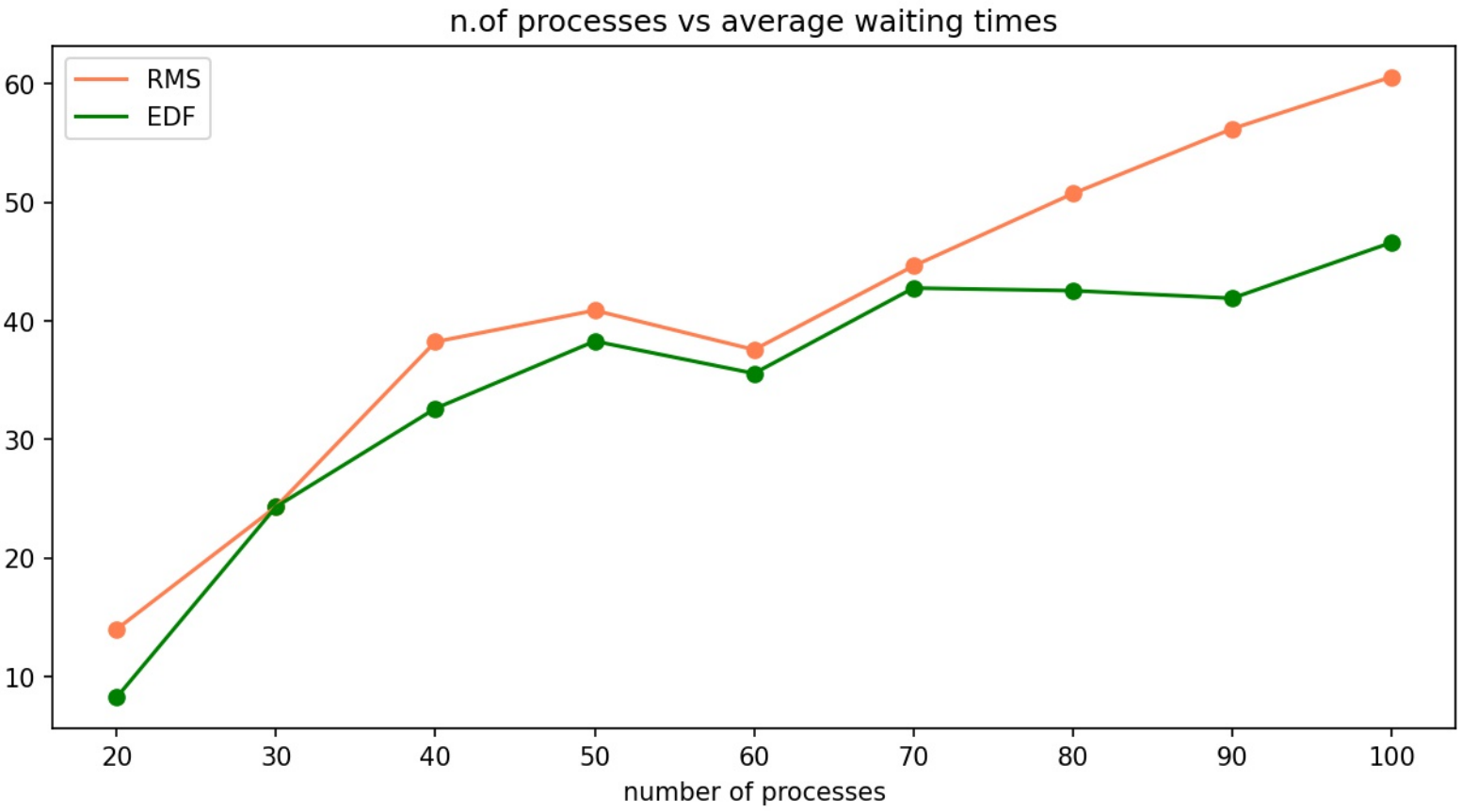
Graphs

fig1 is a overview of number of process that missed the dead line in both algorithms



deadline vs number

fig2 is the overview of average waiting time ***vs*** no of process



time vs number